

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

BAKALÁŘSKÁ PRÁCE



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

**FAKULTA ELEKTROTECHNIKY
A KOMUNIKAČNÍCH TECHNOLOGIÍ**

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

**PROTOTYP SYSTÉMU PRO CENTRALIZOVANÉ
OVLÁDÁNÍ HUDEBNÍCH EFEKTŮ POMOCÍ ANDROID OS**

PROTOTYPE SYSTEM FOR CENTRALIZED CONTROL OF MUSIC EFFECTS USING ANDROID OS

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

Viktor Slezák

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. Vojtěch Zvončák

BRNO 2021

Bakalářská práce

bakalářský studijní program **Audio inženýrství**
specializace Zvuková produkce a nahrávání
Ústav telekomunikací

Student: Viktor Slezák

ID: 203745

Ročník: 3

Akademický rok: 2020/21

NÁZEV TÉMATU:

Prototyp systému pro centralizované ovládání hudebních efektů pomocí Android OS

POKYNY PRO VYPRACOVÁNÍ:

Navrhněte a naprogramujte protokol pro centralizované ovládání parametrů hudebních efektů. Vytvořte a realizujte 3D model akčního členu, který bude jednoduchý na výrobu (malý počet součástí, jednoduché díly). Celý systém realizujte na platformách Arduino a Android. Naprogramujte přehledné grafické uživatelské rozhraní, které umožní přidávat a ovládat hudební efekty a jejich akční členy. Uživatelské rozhraní bude zároveň měnit zobrazované hodnoty v závislosti na fyzickém nastavení parametrů hudebních efektů.

DOPORUČENÁ LITERATURA:

- [1] BREWSTER, David M. Introduction to Guitar Tone & Effects: An Essential Manual for Getting the Best Sounds from Electric Guitars, Amplifiers, Effect Pedals, and Digital Processors. Hal Leonard Corporation, 2003.
- [2] Intro to Eurorack Part 1. Reverb [online]. 2016 [cit. 2019-09-13]. Dostupné z: <https://reverb.com/news/intro-to-eurorack-part-i-doeplers-beginnings-and-power-supply-basics>

Termín zadání: 1.2.2021

Termín odevzdání: 31.5.2021

Vedoucí práce: Ing. Vojtěch Zvončák

doc. Ing. Jiří Schimmel, Ph.D.
předseda rady studijního programu

UPOZORNĚNÍ:

Autor bakalářské práce nesmí při vytváření bakalářské práce porušit autorská práva třetích osob, zejména nesmí zasahovat nedovoleným způsobem do cizích autorských práv osobnostních a musí si být plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č.40/2009 Sb.

ABSTRAKT

Vznik práce byl podnícen myšlenkou ovládat zvukové efekty pohodlně z jednoho místa. Jako příklad této problematiky si můžeme představit kytarový „pedalboard“ ve kterém většinou nalezneme větší počet efektů v podobě malých krabiček s několika otočnými potenciometry. Pokud si kytarista chce změnit nastavení těchto efektů musí se ohnout a jeden po druhém přenastavit. Záměrem této práce je vývoj aplikace a akčního členu pro možnost centrálního ovládání hudebních efektů jedním kliknutím z jednoho místa. Cílem je zjednodušení a centralizování ovládání efektů například kytarových efektů v „pedalboardu“ z jedné aplikace. Pro vývoj se využilo jednodeskového počítače Arduino a přídatného Bluetooth modulu HC-05. Řídící aplikace funguje na platformě Android. V textu je pojednáváno o realizaci aplikace, která je schopná vzdáleně ovládat akční člen na nastavování zvukových efektů, o problematice programování, komunikace přes rozhraní bluetooth a konstrukčním řešením akčního členu s ohledem na tisknutelnost metodou FDM.

KLÍČOVÁ SLOVA

řízení hudebních efektů, arduino, bluetooth, android, java

ABSTRACT

This bachelor thesis had been encouraged by the thought of easily operable sound effects from only one place. Guitar pedalboard can be taken in a consideration as an example, where multiple sound effects are operated by the means of small boxes with several turning potentiometers. When a guitarist wants to change the set up of those sound effects, they must manually reconfigure them separately. The aim of this thesis is therefore development of an application and actuator which enable to operate sound effects from one place by a one-click. The goal is to simplify and centralize the control of sound effects, such are guitar effects in pedalboard, into one application. A single-board computer Arduino and additional Bluetooth module HC-05 are being used for the development. Control systems are run on the Android platform. The text itself deals with the realization of the application, which can remotely control actuator for the sound effects regulation, with the programming problematics, communication via Bluetooth platform and the construction of the solution of actuator with regards to printable method FDM.

KLÍČOVÁ SLOVA

music effect controlling, arduino, Bluetooth, android, java

SLEZÁK, Viktor. *Prototyp systému pro centralizované ovládání hudebních efektů pomocí Android OS*. Brno, 2021. Dostupné také z: <https://www.vutbr.cz/studenti/zav-prace/detail/135541>. Bakalářská práce. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací. Vedoucí práce Ing. Vojtěch Zvončák.

PROHLÁŠENÍ

Prohlašuji, že svou bakalářskou práci na téma „Prototyp systému pro centralizované ovládání hudebních efektů pomocí Android OS“ jsem vypracoval samostatně pod vedením vedoucího práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autor uvedené bakalářské práce dále prohlašuji, že v souvislosti s vytvořením této bakalářské práce jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních nebo majetkových a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

V Brně dne

.....

podpis autora

PODĚKOVÁNÍ

Rád bych poděkoval vedoucímu bakalářské práce, panu Ing. Vojtěchovi Zvončákovi, za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci.

OBSAH

ÚVOD	8
1. Teoretická část	9
1.1. Typy hudebních efektů	9
1.1.2. Kytarové kombo	9
1.1.3. Pedálové efekty (krabičky)	9
1.1.4. Multiefektové processory.....	10
1.1.5. Rackové efekty	11
1.1.6. VST pluginy	11
1.2. Přehled aktuálních řešení.....	12
1.2.1. API 500.....	12
1.2.2. Eurorack.....	13
1.2.3. MIDI	13
1.2.4. Voodoo lab	14
1.2.5. Geckotool	14
REALIZACE.....	15
2. Cíl 1: Návrh akčního členu	15
2.1. Hardware	15
2.1.1. Programovatelný jednodeskový mikropočítač Arduino.....	15
2.1.2. Přídavné Bluetooth moduly.....	16
2.1.3. Servomotory	17
2.1.4. Baterie	18
2.2. Návrh konstrukce	19
2.2.1. Převod.....	19
2.2.2. Uchycení servomotoru a převodu	20
2.2.3. Uchycení na potenciometr	20
2.2.4. Rám.....	21
2.2.5. Krabice pro elektroniku	21
3. Cíl 2: Návrh Aplikace společně s IT infrastrukturou.....	22
3.1. Řídící program Arduina.....	22
3.1.1. Návrh programu	23
3.1.2. Implementace knihoven.....	23
3.1.3. Realizace	23
3.2. Aplikace pro platformu Android.....	24

3.2.1.	Struktura programu	25
3.2.2.	Databáze	26
3.2.3.	Připojení Bluetooth zařízení	27
3.2.4.	Odesílání dat	29
3.2.5.	Třída MainActivity.class	30
3.2.6.	Třída BtControl.class	31
4.	Cíl 3: Konstrukce a testování	32
4.1.	Realizace akčního členu	32
4.1.1.	Realizace uchycení servomotoru a převodu	32
4.1.2.	Rám konstrukce	33
4.2.	Realizace řídicí krabičky	33
4.2.1.	Schéma zapojení	34
4.2.2.	Držák na baterie	35
4.2.3.	Realizace desky plošných spojů	35
5.	Nápady na vylepšení	36
5.3.	Použití BLE	36
5.4.	Vytvoření webové aplikace	36
5.5.	Uchycení akčního členu	36
5.6.	Ovládání na akčním členu	37
5.7.	Ovládání přes MIDI protokol	37
ZÁVĚR		37
SEZNAM ZDROJŮ		37
SEZNAM OBRÁZKŮ		39
SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK		40
SEZNAM PŘÍLOH		41

ÚVOD

V hudebním průmyslu se setkáme s řadou efektů pro úpravu signálu. Tyto efekty mají spoustu podob a řešení, které jsou popsány v kapitole 1.1. Jejich konstrukční řešení se od sebe často liší a není možné je jednoduše a pohodlně ovládat najednou či jejich řízení propojit a centralizovat do jednoho místa. Při průzkumu narazíme na několik existujících řešení popsaných v kapitole 1.2, avšak tyto řešení se převážně soustředí na úzkou část a typ efektů.

Podstatou této bakalářské práce je vymyslet a navrhnout koncept univerzálního, centralizovaného a pohodlného ovládání těchto efektů z jednoho místa a zařízení. Následně potvrdit funkčnost konceptu sestavením prototypu. Díky kterému by se dala ušetřit práce nejen ve studiu. Například doladění zvuku kytarového komba umístěného v nahrávací místnosti přímo z režie. Nebo by kytarista mohl jedním kliknutím vyvolat přednastavené hodnoty pro změnu efektových parametrů celého „pedalboardu“. V rámci bakalářské práce byly stanoveny 3 základní cíle pro otestování funkčnosti konceptu.

První cíl – Návrh konstrukčního řešení akčního členu, který půjde nasadit na co nejširší množství typů hudebních efektů a bude schopen otáčet vybranými potenciometry pro změnu parametrů efektu. V rámci prvního cíle se vybral vhodný hardware s ohledem na jednoduchost a nízkou cenu řešení a následně se vymodeloval akční člen v programu Solidworks tak, aby byl vytisknutelný na 3D tiskárně jako prototyp.

Druhý cíl – Vytvoření IT infrastruktury pro přenos dat mezi akčním členem a řídicí aplikací. V rámci tohoto cíle vznikne mobilní aplikace na platformě Android, která bude uživatelsky jednoduchá a bude umožňovat uživateli snadné ovládání akčního členu a ukládání jednotlivých nastavení pro dané efekty. Bude zde navržen řídicí program pro mikropočítač Arduino, který je schopen přijímat příkazy z mobilní aplikace.

Třetí cíl – Vytvoření prototypu akčního členu a testování jeho funkčnosti v praxi. Zjištění případných slabin konceptu řešení. Proběhne částečné doladění prototypu dle možností 3D tisku. Dojde ke kompletnímu vytisknutí akčního členu a přenosné řídicí krabičky.

Na závěr budou navrženy vhodné vylepšení a řešení pro další vývoj konceptu.

1. Teoretická část

Tato část práce se bude zabývat jednotlivými řešeními dostupných hudebních efektů a momentálních řešení centralizovaného ovládání hudebních efektů.

1.1. Typy hudebních efektů

Již v 50. letech minulého století se začaly objevovat první efekty typu delay řešeny pomocí pásky a elektronek. Vývoj efektu však pokračoval velmi rychle a brzy se začaly objevovat v kytarových kombech například roku 1959 slavné kombo od firmy Vox AC-30 osazeno tak zvaným „Top Boost“ module, s nabízenými efekty vibrato a tremolo. [1] Motivace byla jasná, získat nový a co nejvíce unikátní zvuk. Tento trend pokračuje dodnes a za jejich historii došlo k vývoji mnoha konstrukčních řešení, dále jsou popsány aktuální řešení a realizace efektové úpravy zvukového signálu.

1.1.2. Kytarové kombo

Jako první se každý setká s efekty zabudovanými v zesilovači kytarového komba. Můžeme zde nalézt většinou efekty pro ekvalizaci, zkreslení signálu a přidání prostoru typicky: reverb, boost a další. Tyto efekty je třeba ovládat otáčením potenciometrů přímo na zesilovači, některé řešení dnes umožňují připojit pedál pro přepínání efektů. Neumožňují však ovládání na dálku či ukládání přednastavení.



Obr. 1: Kytarová hlava Vox AC 30 [2]

1.1.3. Pedálové efekty (krabičky)

Pedálové efekty aneb „krabičky“ představují pojem zejména pro kytaristy. Jedná se ve většině případů o krabičku, v níž je zabudován jeden daný typ efektu. „Krabičky“ se mezi sebou mohou libovolně propojovat a vytvářet tak unikátní efektovou cestu daného zvuku. Kytarista si je může poskládat, jak chce, může libovolně efekty měnit a přehazovat. Díky tomu a široké nabídce různých typů efektů, například chorus, flanger, distortion, delay, reverb, phaser, wah-wah, kompresor, ekvalizér, tremolo, vibrato a další, má každý svůj unikátní „pedalboard“ s jedinečným zvukem a spoustou možností.

Bohužel to přináší i své nevýhody. Pokud chce uživatel cokoliv přenastavit, upravit si zvuk kytary nebo pro každou písničku použít jiné nastavení efektů, musí se vždy sklonit a ručně pootáčet potenciometry, dle toho jaké parametry chce změnit. Další nevýhodou je například nemožnost uložit přednastavení daných parametrů a vytváření jejich automatizace.



Obr. 2: Analogový kytarový delay efekt

1.1.4. Multiefektové processory

Multiefekty jsou digitální efekty využívající DPS procesoru pro zpracování zvukových signálů. Tento procesor je určen přímo pro zpracování signálů a díky tomu nedochází k tak velkému zpoždění jako u VST pluginů využívajících CPU počítače popsaných v kapitole 1.1.6. Multiefektové jednotky nám nabízejí velké množství efektů až v řádu desítek. To vše v jedné krabici. Tyto multiefekty jsou ovšem těžší na konstrukci. Čili i jejich pořizovací cena je mnohem vyšší než u analogových efektů. Je to způsobeno právě úpravou signálu v jeho digitální formě. Je zapotřebí signál nejdříve řádně převést na digitální pomocí A/D převodníků, kde při použití levných převodníků dochází ke ztrátě dat a tím pádem kvality signálu. Následně je signál digitálně zpracován a tentokrát pomocí D/A převodníku převeden na analogový signál. Celý proces je tedy mnohem složitější a nákladnější.



Obr. 3: Multiefektový procesor HEADRUSH [3]

1.1.5. Rackové efekty

V rackovém řešení efektů se mohou objevit jak analogové efekty, tak digitální multiefektové procesory. Jedná se pouze o jinak zabudované efekty. Neleží na zemi před kytaristou, ale zabudují se do racku (skříně) například spolu s dalšími efekty. Toto řešení je často používané v nahrávacích studiích nebo u zvukařů při živých koncertech. Jeho omezení je opět to, že je potřeba při změně parametrů fyzicky přijít k efektu a otočit potenciometry.



Obr. 4: Studiový rack s efekty [4]

1.1.6. VST pluginy

Velkou škálu možností nám nabízí digitální efekty v podobě VST pluginů ve spojení s DAW softwarem, tak zvané plug-in moduly. Existuje zde nespočet softwarových „pluginů“, z nichž každý přichází s vlastní jedinečnou funkcí, čímž rozšiřuje celkovou škálu možných zvuků a manipulací. Najdeme zde nepřeberné množství možných efektů distortion, rezonator, reverb, delay, equalizer, syntetizér, kompresor, chorus, virtuální zesilovač, limiter, phaser, flanger a další. Každý z nich má vlastní formu manipulace se zvukem, tónem, výškou a barvou. Chceme-li dosáhnout ještě většího efektu, lze ve vrstvách použít více pluginů a automatizovat jejich řízení. Využití DAW software ovšem není tak jednoduché a vyžaduje mít několik zařízení. Samotný výpočet efektů využívá výpočetního výkonu procesoru počítače, který zároveň zpracovává nespočet jiných úloh. Díky tomuto zaneprázdnění procesoru dochází ke zpoždění výstupního signálu. Délka zpoždění závisí na výpočetním výkonu procesoru.



Obr. 5: Ukázka VST pluginu RoomWorks SE v DAW softwaru Cubase

1.2. Přehled aktuálních řešení

Dnes máme dostupné velké množství analogových i digitálních efektů i multiektů. Jednoduchých způsobů, jak tyto efekty ovládat vzdáleně není mnoho. Existují způsoby úpravy signálu přes počítač či tablet za pomoci zvukové karty, multiefekty s DSP procesorem, analogové efekty s digitálním řízením, či analogové efekty s otočnými potenciometry. Jejich seskupení a centrální ovládání je v některých případech jednoduché například u kytarových multiektů, kdy jeden multiekt obsahuje desítky různých efektů v jednom zařízení. Ovšem analogové efekty se vyrábí po jednotlivých krabičkách (jedno zařízení jeden efekt). Efekty se následně propojují mezi sebou a jejich centralizované ovládání není možné. Pomocí rozhraní popisovaného v této práci by bylo možné řídit všechny typy těchto efektů z jediného místa a zařízení. V této části jsou popsány existující řešení centralizovaného ovládání, většina je ovšem velmi úzce zaměřená nebo finančně nákladná.

1.2.1. API 500

Rozhraní API 500 je standardizovaný rack do kterého můžeme libovolně vkládat efektové moduly dle našich potřeb a možností. Na trhu dnes najdeme různé analogové moduly. Často se do něj vkládá předzesilovač a následné moduly pro úpravu signálu. Moduly jsou vždy analogové a najdeme zde kopie legendárních efektů typu Pultec nebo Neve od značek Warm audio, Elysia, Heritage audio a další. Rozhraní vzniklo díky unikátnímu zesilovacímu tranzistoru API 2520, který byl specifický svým zvukem a který dal vlastně obsah tomu čemu dnes říkáme „americký zvuk“ - to znamená detailní barevný zvuk s výraznými vyššími středy. Staly se evergreenem historie nahrávací techniky.

Začátek osmdesátých let si audio inženýři začali stavět své vlastní neoficiální racky, do kterých zabudovávali různé vintage API moduly. Značně používaným se stal čtyřmodulový API 500 rack do kterého inženýr Art Blem vložil předzesilovač, kompresor, ekvalizér a vytvořil tím modulární chanelstrip, posléze nazývaný jako „Lunchbox“. [5] V roce 2010 se stal formát API 500 standardem. Rozhraní je ovšem velmi drahé a pořízení 4 modulového racku s moduly může vyjít na několik desítek tisíc korun.

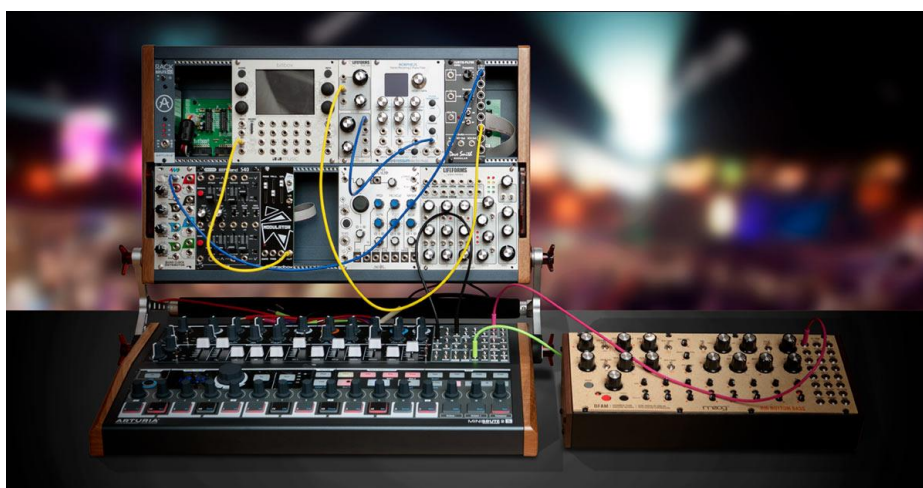


Obr. 6: Api lunchbox [6]

1.2.2. Eurorack

Řešení Eurorack vzniklo v roce 1996 od tvůrce Dieter Doepfer a je velmi podobné jako rozhraní API500. Jde o modulární skládání z řady různých efektů, syntezátorů a dalších modulů do jednoho balíčku (racku). Moduly zde mezi sebou komunikují a můžeme jejich vzájemným propojováním vytvářet jedinečný zvukový projev.

Jedná se o standardizovaný rack 3U osazený společným napájením a sběrnici pro propojení modulů do které jsou zapojovány pomocí 16 pinového konektoru ve kterém je obsáhlé nejen napájení, ale také části pro společnou komunikaci a synchronizaci. [7] Eurorack je však zaměřen spíše na tvorbu unikátních a snadno měnitelných syntezátorů. Co se týče ovládání opět je třeba veškeré parametry měnit ručním otáčením potenciometrů na daných modulech.



Obr. 7: Eurorack [8]

1.2.3. MIDI

V dnešní době velmi rozšířený MIDI protokol pro ovládání všech druhů hudebních zařízení. Ve velké míře použit pro přenos a zápis tónů například z kláves, kde danému zápisu tónů můžeme přiřadit libovolné banky zvuků. Protokol je velmi jednoduchý a efektivní. Proto se rozšířil i pro ovládání parametrů hudebních efektů, zejména těch digitálních. Například MIDI kontroler k ovládání parametrů DAW systémů. Podobným způsobem lze řídit i efekty rackové a někdy i krabičky. Problémem je však, že efekty musí mít tyto možnosti implementovány a mít vstup pro MIDI příkazy. Toto opatření může být ve výsledku velmi nákladné a zdaleka není univerzální.



Obr. 8: MIDI kontroler [9]

1.2.4. Voodoo lab

Řešení od společnosti Voodoo lab se zaměřuje zejména na kytaristy a jejich pedlboardy kdy nabízejí kompletní možnosti pro sestavení vlastního pedalboardu. Řešení je realizováno přepínači. Uživatel si zapojí efektové krabičky do různých smyček a tyto smyčky může následně mezi sebou přepínat a kombinovat. Smyčky je také možné řídit pomocí MIDI protokolu a kombinovat tak i s dalšími MIDI efekty [10].

Společnost vytváří zajímavé propojení krabičkových efektů a efektů řízenými pomocí MIDI protokolu. Není však možné z jednoho místa upravovat přímo parametry jednotlivých efektů nebo je kombinovat s dalšími typy efektů jako jsou například multiefekty.

1.2.5. Geckotool

V průběhu práce na bakalářské práci byl objeven projekt Geckotool na Kickstarteru, který se zabývá stejnou problematikou centralizovaného ovládání hudebních efektů. Problém je řešen velmi podobným a inspirativním způsobem. Řešení se skládá z hlavní řídicí krabičky, do které se zapojí moduly. Tyto moduly následně nasadíme na potenciometr hudebního efektu a připevníme aretační maticí, aby se nehýbaly. Dané potenciometry následně můžeme ovládat přes MIDI protokol, automatizovat průběh jejich změny nebo je ovládat přes mobilní aplikaci. [11]

Nevýhodou je, že moduly, které nasadíme na potenciometry efektu jsou příliš velké. Pokud máme na efektu potenciometry příliš blízko, nepodaří se nám moduly nasadit na všechny potenciometry. Řešení je to povedené, ale drahé. Základní cena za řídicí krabičku a 3 moduly se pohybuje v přepočtu kolem 7000,- Kč.



Obr. 9: Geckotool [11]

REALIZACE

2. Cíl 1: Návrh akčního členu

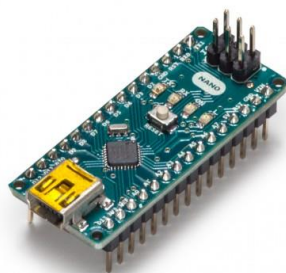
Prvním cílem je navrhnout a vymodelovat akční člen v programu Solidworks, tak aby byl následný prototyp vytisknutelný na 3D tiskárně metodou MDF tisku a zároveň, aby byla před tiskem ověřena jeho funkčnost a odhalily se problémová místa. Také je zde popsán výběr hardwaru, který akční člen nakonec rozpohybuje.

2.1. Hardware

Před tím, než se začne modelovat akční člen je potřeba zvolit vhodný hardware, abychom pro něj dokázali model uzpůsobit. Jsou položeny otázky. Co bude otáčet potenciometry? Jakým způsobem bude probíhat komunikace s aplikací? Co bude zpracovávat příkazy z aplikace a přenášet do fyzických pohybů? Jakým způsobem bude akční člen napájen?

2.1.1. Programovatelný jednodeskový mikropočítač Arduino.

Jako řídicí mikropočítač byl již v zadání zvoleno Arduino, pro naše potřeby v co nejmenší velikosti byl vybrán model Arduino NANO. Arduino je otevřená elektronická platforma, založená na jednoduché počítačové desce a vývojovém prostředí. Pomocí Arduina můžeme vytvářet interaktivní objekty. Arduino deska získává údaje od snímačů a senzorů, na základě získaných údajů ovládá výstupy. [12] Mikropočítač je možné jednoduše rozšířit o přídatné moduly. Celé řešení je jednoduché levné a má nízkou spotřebu energie.



Obr. 10: Arduino NANO [13]

Technické parametry mikropočítače Arduino NANO:

Mikroprocesor:	Atmega328
Pracovní napětí:	5 V
Napájecí napětí:	6-12 V
Spotřeba:	19 mA
Analogové piny:	9
Digitální piny:	13 (z toho 6 PWM)
PWM výstupů:	6
Rozměry:	18 x 45 mm

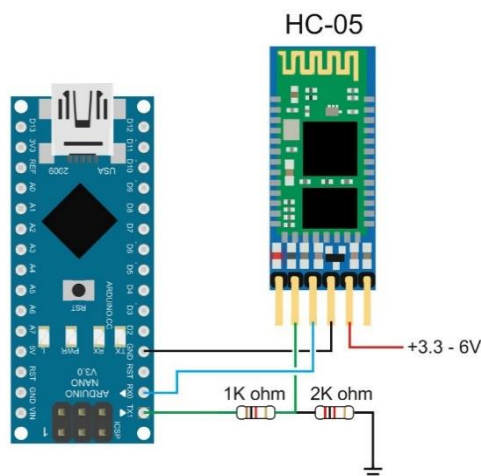
2.1.2. Přídavné Bluetooth moduly

Protože bude potřeba komunikovat s řídicí aplikací probíhající na jiné platformě, je potřeba k Arduino přidat modul pro bezdrátovou komunikaci. V našem případě se zvolilo rozhraní Bluetooth pro jeho dostupnost a jednoduchou práci s komunikačním protokolem. Jednodeskový počítač Arduino poskytuje jednoduché připojení Bluetooth modulu. Zvolil se konkrétně modul HC-05, který podporuje protokol Bluetooth classic verze 2.0 a je schopen s Arduinem komunikovat přes sériové rozhraní. Protokol verze 2.0 je sice starší, ale pro potřeby práce dostačující.

Připojení Bluetooth modulu HC-05 je jednoduché. Modul má napájení 3,6 – 6 V. V našem případě se připojí k výstupnímu napájecímu pinu desky, který má na výstupu napětí 5 V. Zem připojíme na společnou zem desky plošných spojů. Piny RXD a TXD slouží pro sériovou komunikaci UART. Pin RXD pro čtení dat a pin TXD pro posílání dat. V našem případě se tyto piny připojí na piny 0 a 1 desky Arduino. Pin TXD na pin 0 a pin RXD na pin 1. Na desce si můžeme všimnout, že vstupy 1 a 0 jsou již připraveny pro sériovou komunikaci. Protože bluetooth modul HC-05 pracuje s vnitřním napětím 3,6 V a zvolené Arduino NANO má výstupní napětí 5 V je potřeba v komunikační cestě mezi pinem 1, který slouží pro odesílání dat a pinem RXD na modulu, který slouží pro přijímání dat, vložit odporový dělič, abychom modul nezatěžovali. [14] Schéma zapojení je na obr. 11.

Technické parametry bluetooth modulu HC-05:

Napájecí napětí:	3,6 – 6 V
Odběr proudu:	30-40 mA při párování (10 mA spárovaný)
Komunikace:	UART (RXD, TXD)
Výkon:	Class 2 (cca 4 dBm / 2,5 mW)
Dosah:	cca 10 m
Bluetooth specifikace:	v2.0 + EDR



Obr. 11: Zapojení Bluetooth modulu HC-05 [14]

Výpočet odporového děliče:

Modul má na komunikačním pinu vysoký vstupní odpor, počítáme tedy odporový dělič jako nezatížený. R_1 volíme 1 k Ω

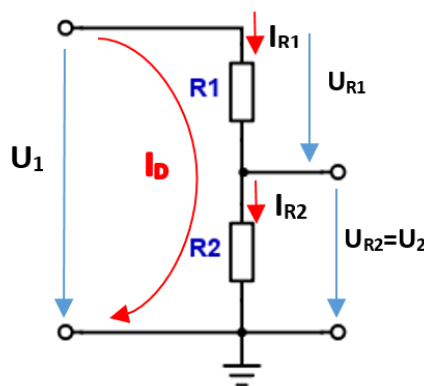
$$U_1 = 5\text{ V}, U_2 = 3,6\text{ V}, R_1 \text{ volíme } 1\text{ k}\Omega$$

$$I_1 = \frac{U_{R1}}{R_1} = \frac{U_1 - U_2}{R_1} = \frac{5 - 3,6}{1000} = 1,4\text{ mA} = I$$

$$U_2 = I * R_2$$

$$R_2 = \frac{U_2}{I} = \frac{3,6}{0,0014} = 2571\ \Omega$$

Volíme nejbližší nižší: $R_2 = 2,2\text{ k}\Omega$

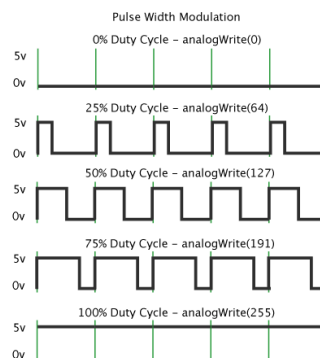


Obr. 12: Návrh nezatíženého děliče-výpočet rezistorů [15]

2.1.3. Servomotory

Jako pohonná jednotka se zvolil servomotor, který umožňuje dostatečně přesné řízení úhlu natočení. Byl zvolen konkrétně servomotor Tower Pro SG90 který má dostatečnou sílu, malou velikost a nízkou pořizovací cenu. Servomotor je řízen pomocí PWM signálu, který Arduino zvládá jednoduše generovat na některých pinech.

PWM modulace funguje střídáním sepnutého signálu a vypnutého signálu v jedné periodě. Tomuto poměru se nazývá střída signálu a udává se v procentech. Střídání stavu vypnuto a zapnuto v různých střídách má za následek změnu efektivní hodnoty napětí signálu. Díky tomu můžeme ovládat otáčky motoru nebo jas diody. Tohoto principu využívá také ovládání servomotorů. [16]



Obr. 13: Pulzně šířková modulace [16]

Servomotor má 3 vstupní piny na konektoru 2 napájení, které může být v rozsahu napětí 3,0 – 7,2 V. Třetí vstup je pro řídicí signál PWM modulace, kdy podle střídy modulace dojde k danému natočení servomotoru.

Technické parametry servomotoru SG90:

Provozní napětí:	3,0 – 7,2 V
Odběr proudu v zátěži:	550 mA
Rychlost:	0,12 s / 60° (při 5 V)
Točivý moment:	1,6 kg (při 6 V)
Rozměry:	23 x 23 x 12 mm



Obr. 14: Servo SG90 [17]

2.1.4. Baterie

Aby bylo docíleno univerzálního a jednoduchého použití akčního členu kdekoliv, je potřeba akční člen napájet z baterie. Standardní baterie typu AA například NiMH mají napětí 1,2 V. Potřebujeme napájecí napětí v rozmezí 6-7 V. Kvůli velké spotřebě servomotorů v zátěži 550 mA s rezervou počítáno 750 mA na jeden servomotor potřebujeme baterie, které jsou schopny dodávat proud až 2,5 A s minimálním poklesem napětí.

Proto byly zvoleny baterie typu Li-ion, které disponují velkou kapacitou a jsou schopny dodat rychle velký proud až 30 A. Jejich napětí je 3,6 V. Díky tomu stačí sériové spojení dvou baterií. Zvolený typ baterie Sony US18650VTC5 Li-ion 3,6 V 2600 mAh. Jedinou nevýhodou baterie je vyšší pořizovací cena 270,- Kč / ks a potřeba speciální nabíječky.

Odhadovaná doba výdrže akčního členu bez nabití je 2 h při konstantním zatížení servomotorů. V klidovém režimu se dostáváme na desítky hodin. Tato výdrž je pro prototyp dostačující.

Technické parametry baterie Sony US18650VTC5:

Typ:	Li-Ion
Kapacita:	2600 mAh
Napětí:	3,6 V
Maximální proud:	30 A
Rozměry:	18,2 x 18,2 x 62 mm



Obr. 15: Li-ion baterie

2.2. Návrh konstrukce

Návrh konstrukce bude realizován jako 3D model v programu Solidworks. Cílem je navrhnout rám, který tvoří oblast, ve které se mohou pohybovat servomotory nahoru a dolů. Servomotory tak mohou být kdekoliv uvnitř rámu připevněny k otočným potenciometrům hudebního efektu.

Pro dokázání funkčnosti konceptu bylo zvoleno použití 3 servomotorů na jeden akční člen. Využilo se vlastnosti servomotorů, že dokud je na jejich řídicí vstup přiváděn PWM signál, tak drží úhel natočení odpovídající střídě signálu. Pokud jsou tedy připevněny 3 servomotory v jednom rámu, samy si navzájem zabrání otáčení servomotoru a rámu namísto potenciometrů. Tento jev nemusí fungovat, pokud se otáčejí všechny servomotory najednou stejným směrem. Návrhy na vylepšení pro reálné použití jsou popsány níže v kapitole 5.

2.2.1. Převod

Protože servomotory se mohou otáčet pouze v rozmezí $0^\circ - 180^\circ$ je potřeba navrhnout jednoduchý převod, abychom docílili většího úhlu natočení. Dle průzkumu je možné ve většině případů potenciometrem u hudebních efektů otáčet v rozmezí $0^\circ - 270^\circ$. Požadovaný úhel natočení je zvolen s rezervou na 360° . Díky velkému točivému momentu servomotoru to je možné.

Převodový poměr je zvolen 0,5 a počet zubů hnacího kola je zvolen 30.

$$i = 0,5, Z_1 = 30 \text{ zubů}$$

$$i = \frac{Z_2}{Z_1} = \frac{D_2}{D_1} \quad [18]$$

$$Z_2 = Z_1 * i = 30 * 0,5 = 15 \text{ zubů}$$

Protože je potřeba ozubená kola co nejmenší, ale zároveň vytisknutelné na 3D tiskárně volíme modul 0,3 mm.

$$m = 0,3 \text{ mm}$$

$$m = \frac{D}{Z} \quad [19]$$

$$D_1 = Z_1 * m = 30 * 0,3 = 9 \text{ mm}$$

$$D_2 = Z_2 * m = 15 * 0,3 = 4,5 \text{ mm}$$

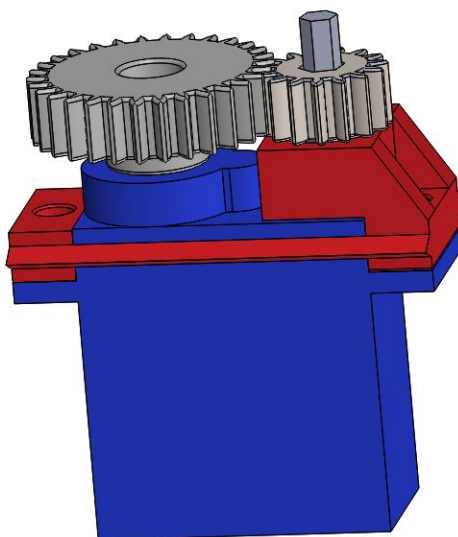
Nyní jsou známy potřebné parametry pro použití nástroje Tool box integrovaného v programu Solidworks. Tento nástroj je schopen vygenerovat ozubená kola podle norem ISO.

2.2.2. Uchytení servomotoru a převodu

Větší hnané kolo bude nasazeno přímo na výstupní ozubené kolo servomotoru, které má ve středu závit, takže se aretuje šroubkem. Menší hnací ozubené kolo je potřeba usadit. Proto se navrhl držák viz obr. 16, který se přišroubuje na kraje servomotoru a na horní části se osadí ložisko pro hřídel na které bude hnací ozubené kolo a na druhé straně bude přes nástavec připojen k potenciometru.

Funkčnost převodu ozubenými koly závisí na jejich upevnění. Hřídel pro hnací ozubené kolo je upevněna v ložisku pouze na jedné straně a může docházet k vyhnutí hřídele a přeskokování zubů v převodu. Proto byl držák upraven a přidalo se druhé ložisko tak, aby bylo hnací ozubené kolo co nejblíže ložisku. Díky tomu nedochází k tak velkému vyhnutí z osy hřídele.

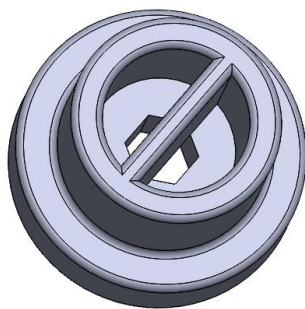
Držák na ozubené kolo slouží zároveň pro pojezd v drážce rámu. Tvar pojezdu byl upraven do trojúhelníku pro snadné tisknutí bez podpěr.



Obr. 16: Uchytení servomotoru

2.2.3. Uchytení na potenciometr

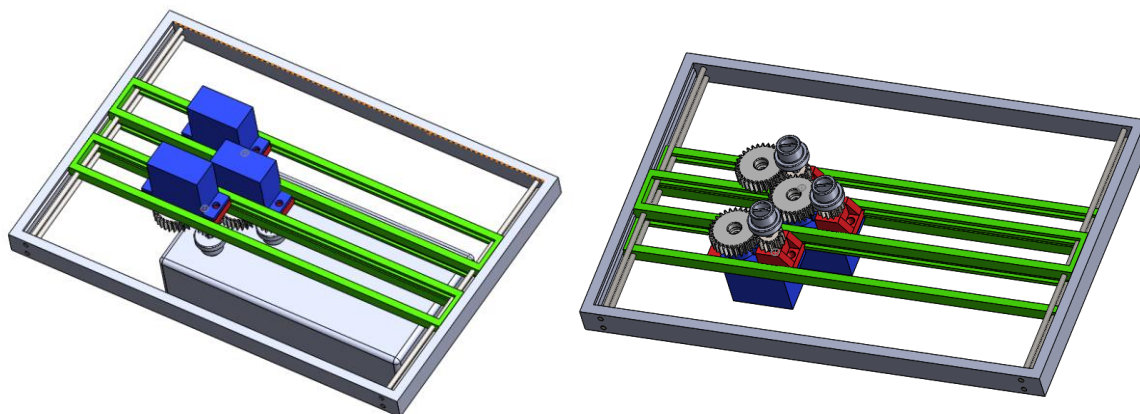
Při návrhu adaptéru pro uchycení akčního členu k potenciometru bylo využito faktu, že potenciometry se vyrábějí normovaně, takže navrhnout univerzální adaptér je velmi jednoduché. Na jedné straně ho nasadíme na hřídel z převodu a na straně druhé nasuneme na potenciometr. Uprostřed je drážka abychom zajistili, že se nebude protáčet viz obr. 17.



Obr. 17: Adaptér na potenciometr

2.2.4. Rám

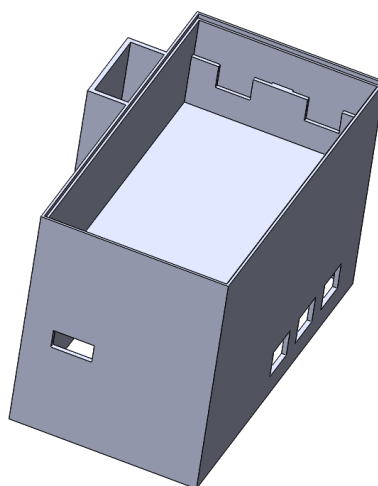
Aby se servomotory neotáčely kolem osy potenciometru je potřeba je upevnit do rámu. Byly zvoleny 2 pojezdy pro osy x a y. V rámci osy y se pohybuje servomotor v pojezdu servomotoru. Tento pojezd je následně upevněn v drážkách rámu viz obr. 18 a 19. Toto řešení má slabé místo. Pokud se pojezd v rámu nepohybuje na obou stranách zároveň, vypadne z drážek, ve kterých je upevněn. Proto byly do rámu přidány pojezdové hlazené tyče, na kterých pojezd drží viz obr. 18 a 19.



Obr. 18 a 19: Sestava akčního členu

2.2.5. Krabíčka pro elektroniku

Na krabíčku byl požadavek jediný, ať je malá a jednoduchá. V rámci konstrukce krabíčky je z vrchní strany držák na baterie a ve spodní části je místo pro desku plošných spojů, spolu s Bluetooth modulem, vypínačem a konektory pro připojení 3 servomotorů. Z boku krabíčky nalezneme tunel pro Bluetooth modul, aby jeho anténa trčela nahoru mimo krabíčku. Díky tomu se zajistí větší dosah Bluetooth signálu.



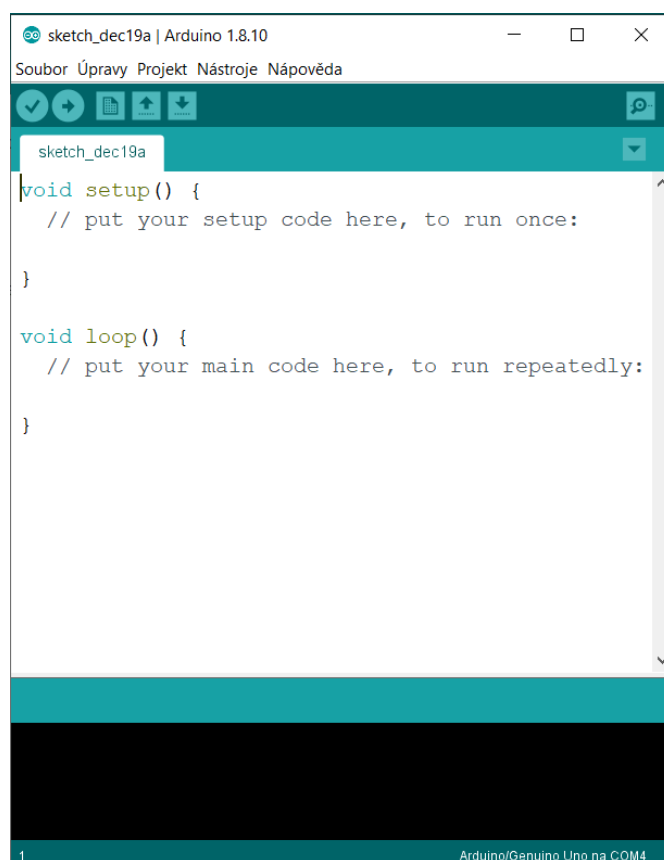
Obr. 20: Krabíčka pro elektroniku

3. Cíl 2: Návrh Aplikace společně s IT infrastrukturou

3.1. Řídící program Arduina

Řídící program pro arduino je psán ve freeware vývojovém prostředí Arduino IDE dodávaném přím pro mikropočítače Arduino. Mikropočítač arduino se programuje pomocí jazyku C++. Vývojové prostředí má již při založení nového projektu implementovanou knihovnu Wiring, která nabízí jednoduché funkce pro programování mikropočítače Arduino.

Při vygenerování nového programu se automaticky vytvoří dvě základní funkce, které považujeme za základ celého programu a do kterých píšeme náš kód. Jedná se o funkce *setup ()* a *loop ()*. Tyto funkce můžeme vidět na obr. 21.

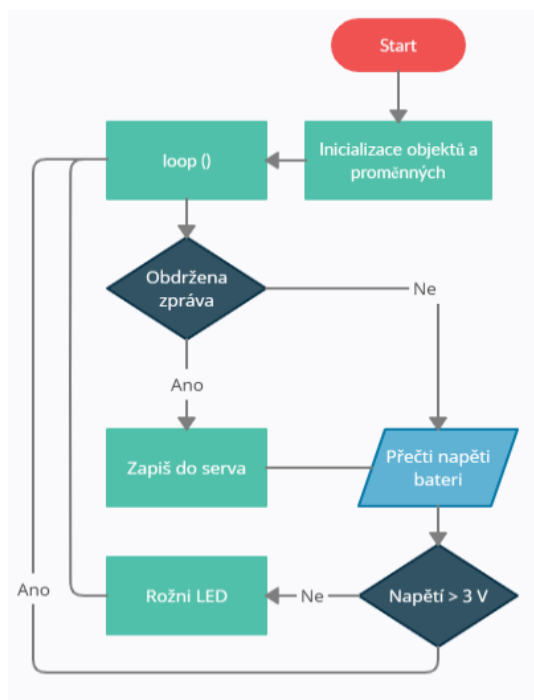


Obr. 21: Vývojové prostředí Arduino IDE

3.1.1. Návrh programu

Od programu pro Arduino je požadováno zejména zpracování obdržených řídicích pokynů z Bluetooth modulu a následné nastavení servomotorů dle obdržených pokynů. Potřebujeme tedy smyčku, která bude naslouchat sériové komunikaci a pokud obdrží nová data nastaví podle nich daný servomotor.

V konstrukci jsou použity baterie typu Li-Ion, které se nesmějí nikdy vybit pod kritickou hranici, jež byla v rámci této práce zvýšena na 3 V, jinak by došlo k jejich znehodnocení. Je potřeba, aby Arduino hlídalo a odečítalo úroveň napětí na bateriích a pokud bude překročena kritická úroveň rozsvítí se kontrolní dioda.



Obr. 22: Blokový diagram programu pro Arduino

3.1.2. Implementace knihoven

Nejprve je potřeba do vývojového prostředí vložit rozšiřující knihovnu s již vytvořenými funkcemi, které velmi usnadní ovládání servomotorů. Využije se knihovna `Servo.h` [20].

Knihovna `Servo.h`, jak vypovídá její název, nám umožní pracovat se servomotory. Knihovna nám nabízí funkce pro jednoduché a snadné ovládání servomotorů. Servomotory následně řídíme funkcí `Servo.write()`, kde můžeme zapsat hodnoty od 0-180°. Dle toho se servomotor natočí. Ovládání servomotorů funguje na principu PWM, která byla popsána v kapitole 2.1.3 a knihovna za nás mapuje rozmezí dostupné PWM modulace na jednoduchý zápis ve stupních.

3.1.3. Realizace

V prvním kroku inicializujeme objekty a proměnné které jsou následně potřeba. První vytvoříme objekt `Servo` pro každý servomotor a přiřadíme je na digitální piny 4, 8 a 12.

Dále jsou potřeba proměnné typu `Integer` a to **data** pro uložení přijaté hodnoty z Bluetooth modulu, **batteryLowLED** pro označení digitálního výstupu č. 10 ke kterému je připojena led dioda indikující vybitou baterii, **batteryStateInput** pro označení analogového vstupu č. A1 pro čtení napětí baterie a proměnou **batteryState** do které je uloženo naměřené napětí baterie.

Protože máme 3 servomotory a je potřebné, aby komunikace mezi mobilní aplikací a Arduinem byla co nejjednodušší, kvůli rychlosti přenosu, tak se posílají hodnoty 1-543. Následně pomocí `if else` příkazu jsou rozděleny pro dané servomotory. Servomotoru 1 odpovídají hodnoty 1-181, servomotoru 2

hodnoty 182–362 a servomotoru 3 hodnoty 363–543. Dle daného servomotoru je následně od čísla odečtena hodnota 1, 182 nebo 363 abychom dosáhli výstupu 0-180 který je následně zapsán do daného servomotoru, viz ukázka kódu níže. Po zápisu do servomotoru je program vždy pozastaven na 5 ms aby nedocházelo k zahlcení servomotorů.

```
If (Serial.available() > 0){  
    data = null;  
    data = Serial.parseInt();  
  
    if (data <= 181 && data > 0){  
        mServo_01.write(data - 1);  
  
        delay(5);  
  
    } else if (data <= 362 && data > 181){  
        mServo_02.write(data - 182);  
  
        delay(5);  
  
    } else if (data <= 543 && data > 362){  
        mServo_03.write(data-363);  
  
        delay(5);  
    }  
}
```

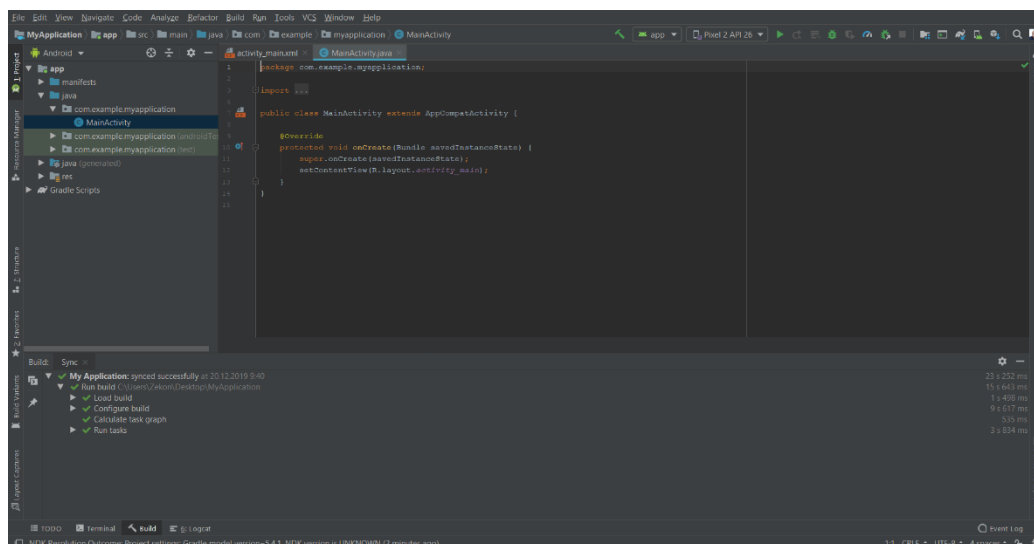
Jako poslední proběhne kontrola stavu baterie při které se z portu A1 odčte napětí na baterii pomocí příkazu analogRead(), kdy je pomocí 10 bitového A/D převodníku porovnána hodnota napětí na vstupu s referenční hodnotou u mikropočítače Arduino NANO v základním nastavení 5 V. Funkce vrátí hodnotu 0-1023 kdy 0 značí 0 V na vstupu a 1023 značí 5 V na vstupu. To znamená, že jedna jednotka = 0,005 V => 3 V = 3 / 0,005 = 600. Na základě tohoto výpočtu zadáme pomocí příkazu if kontrolu, jestliže odečtené na baterii je menší než 600 zapíšeme rozsvítíme kontrolní led diodu viz ukázka kódu níže.

```
batteryState = analogRead(batteryStateInput);  
if (batteryState < 600){  
    digitalWrite(batteryLowLED, HIGH);  
}
```

3.2. Aplikace pro platformu Android

Mobilní aplikace pro platformu Android bude psána ve vývojovém prostředí Android studio, které je dostupné zdarma od společnosti Google. Android Studio umožňuje psát aplikaci ve dvou jazycích a to v jazyce Java nebo Kotlin. V našem případě byl vybrán jazyk Java.

Android studio nabízí nespočet možností pro vývoj aplikací na platformu Android a jedná se o nejpoužívanější vývojové prostředí pro vývoj mobilních aplikací. Jeho vývoj spadá pod společnost Google, která vyvíjí také operační systém Android. Vývojové prostředí nabízí možnosti ladění aplikace, kdy můžeme po připojení telefonu k PC vidět grafy v reálném čase pro vytížení procesoru spotřebu baterie využití paměti a další. Ve studiu nalezneme také grafický editor. Díky němu nemusíme veškerý vizuál aplikace psát kódem, ale je možné ho tvořit rovnou přidáváním bloků na plochu, nebo psát kód a zároveň kontrolovat vizuální výsledek.



Obr. 23: Vývojové prostředí Android studio

3.2.1. Struktura programu

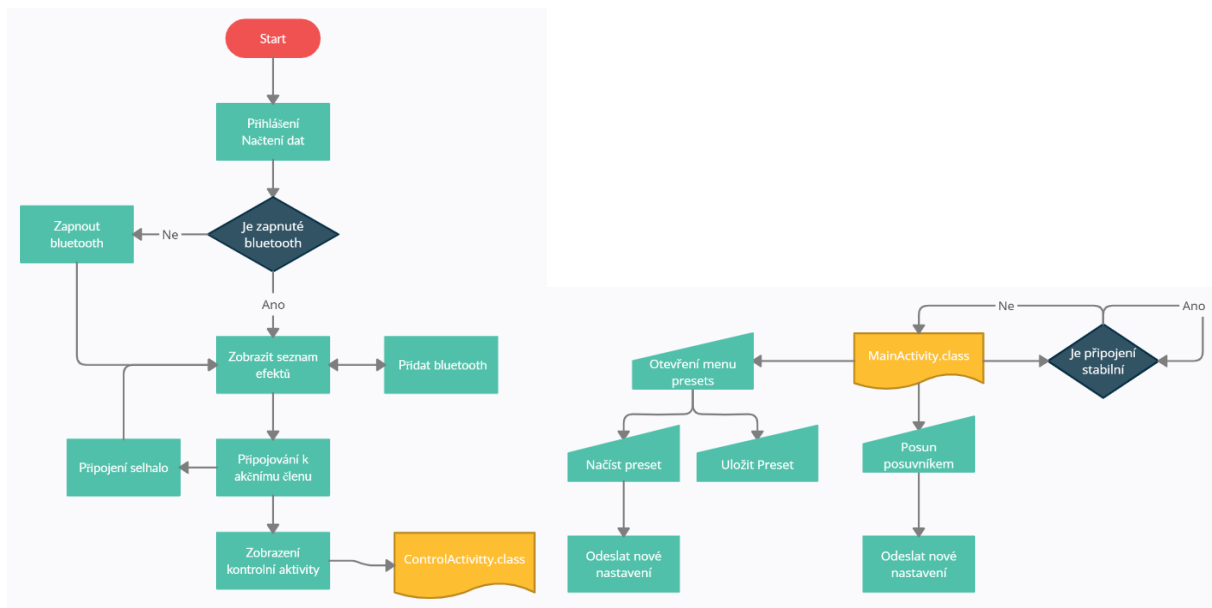
Protože je požadováno univerzální řešení bylo zvoleno přidat možnost ukládání dat o nastavení efektů a jejich parametrů a přednastavení na server. Za tímto účelem je využito služby Firebase, více o službě v části 3.2.2. Díky tomu je potřeba jako první zajistit registraci uživatele, přihlášení do aplikace a funkce pro změnu hesla při jeho zapomenutí.

Dále je potřeba, aby uživatel mohl jednoduše přidávat další akční členy do své databáze efektů. Za tímto účelem budou vytvořeny dvě možnosti. První je zobrazení seznamu spárovaných Bluetooth zařízení a výběr z tohoto seznamu. Nebo naskenování QR kódu pomocí integrované čtečky v aplikaci. QR kód je přilepen na každém akčním členu a obsahuje unikátní Bluetooth MAC adresu daného modulu. Po výběru nebo naskenování akčního členu uživatel zadá jméno daného efektu a je přidán do jeho seznamu efektů.

V seznamu efektů může uživatel mazat a přejmenovávat dané efekty. Po výběru konkrétního efektu je zahájeno připojování k danému akčnímu členu. Po úspěšném připojení se zobrazí kontrolní aktivita.

Kontrolní aktivita má za úkol veškeré ovládání akčního členu. Pokud se uživatel připojí k danému akčnímu členu poprvé, je potřeba nastavit limity efektů proto slouží vyskakovací dialogová okna, které uživatele provedou potřebným nastavením. Následně může pomocí táhel ovládat parametry efektu. Dále zde jde uložit přednastavení parametrů a následně je vyvolávat. To je realizované opět pomocí dialogového okna se seznamem uložených přednastavení.

Po návratu na seznam efektů dojde automaticky k odpojení od daného akčního členu z důvodu šetření energie, protože použitý typ Bluetooth připojení je náročnější na spotřebu energie, viz popis Bluetooth modulu v kapitole 2.1.2.



Obr. 24 a 25: Blokové schéma aplikace pro Android

3.2.2. Databáze

Díky databázi bude moci uživatel ke svým uloženým efektům a presetům přistupovat z jakéhokoliv mobilního zařízení.

Pro zajištění funkční databáze bylo využito řešení od společnosti Firebase, která umožňuje implementovat databázi jednoduše na všech platformách. Pro tuto práci jsou důležité platformy zejména Android OS a web [21].

Je využito služeb Firebase Authentication, která nám umožňuje jednoduchou správu uživatelů, jejich registraci, přihlášení a správu hesla. Každý uživatel má v databázi svoje unikátní identifikační číslo.

Dále je použita služba Cloud Firestore, která nám umožňuje ukládat data a objekty. V Našem případě ukládáme objekt User označený vždy unikátním uživatelským identifikačním čísle. Tento objekt následně obsahuje pole objektů Efekt. A každý objekt efekt obsahuje informace o nastavení efektu a pole objektů preset.

Tímto způsobem jsou uloženy veškerá data uživatele. Díky tomu je možné k nim přistupovat odkudkoliv a je zajištěna jednoduchá rozšiřitelnost na jiné platformy.

Ukázka funkce logIn():

```
private void logIn() {
    buttonLogIn.setOnClickListener(v -> {
        if (editTextEmail.getText().toString().equals("")) {
            Toast.makeText(this, "Can not Log-in without email",
                Toast.LENGTH_SHORT).show();
        } else if (editTextPassword.getText().toString().equals("")) {
            Toast.makeText(this, "Can not Log-in without password.",
                Toast.LENGTH_SHORT).show();
        } else {

            mAuth.signInWithEmailAndPassword(editTextEmail.getText().toString(),
                editTextPassword.getText().toString()).addOnCompleteListener(this, task ->
            {
                if (task.isSuccessful()) {
                    Log.d(TAG, "logIn: success");
                    FirebaseUser currentUser = mAuth.getCurrentUser();
                    startMainActivity(currentUser);
                } else {
                    Log.d(TAG, "logIn: failure", task.getException());
                    Toast.makeText(this, "Log-in failed.",
                        Toast.LENGTH_SHORT).show();
                }
            });
        }
    });
}
```

Ukázka funkce loadData():

```
private void loadData() {
    database.collection("users").document(currentUser.getId()).get()
        .addOnSuccessListener(documentSnapshot -> {
            user = documentSnapshot.toObject(User.class);
            ourDeviceList = user.getOurDeviceList();
            buildOurDeviceListView();

        }).addOnFailureListener(e -> {
            user = new User(new ArrayList<>());
            ourDeviceList = new ArrayList<>();
            buildOurDeviceListView();
        });
}
```

3.2.3. Připojení Bluetooth zařízení

V této práci je použito Bluetooth classic v2.0 u kterého je možné realizovat připojení dvěma způsoby, jako klient nebo jako server. V našem případě se zvolilo připojení jako klient, kdy modul Bluetooth HC-05 vytváří server a zařízení se k němu snaží připojit. Pro připojení je důležité UUID, které má každý server jako své unikátní ID. V případě sériově vyráběných Bluetooth modulů bývá standardně: 00001101-0000-1000-8000-00805F9B34FB.

Při metodě připojení mobilního zařízení k Bluetooth modulu jako klient se následně modul stává serverem a funguje na principu neustálého naslouchání a čekání na požadavek klienta (mobilu).

Zařízení vytváří BluetoothServerSocket a vyčkává na požadavek klienta o přihlášení. Po tomto požadavku zařízení potvrdí BluetoothSocket a dojde ke zrušení BluetoothServerSocket, aby nedošlo k žádosti o připojení od dalšího zařízení, které by mohlo zrušit stávající připojení. [22]

Před zahájením připojení k Bluetooth modulu je nejprve potřeba modul vyhledat a zkontrolovat, jestli má uživatel na mobilním zařízení zapnutý Bluetooth přijímač. Kontrola, zdali je na mobilním zařízení zapnuté Bluetooth proběhne ihned při přihlášení uživatele do aplikace. V případě že není zapnuté zobrazí se dialogové okno s tlačítkem pro zapnutí Bluetooth.

Ukázka funkce isBluetoothEnable():

```
public void isBluetoothEnable() {
    if (!BluetoothAdapter.isEnabled()) {
        Intent enableBtIntent = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
        try {
            startActivityForResult(enableBtIntent, REQUEST_ENABLE_BT);
        } catch (Exception e) {
            Toast.makeText(MainActivity.this, "Bluetooth failed",
Toast.LENGTH_SHORT).show();
        }
    }
}
```

Než se program pokusí o připojení k Bluetooth modulem je nejprve potřeba získat jeho bluetooth MAC adresu. Pomocí MAC adresy dochází k navázání spojení s modulem. Uživatel má dvě možnosti vyhledání dostupných akčních členů. Může zobrazit seznam spárovaných zařízení nebo naskenovat QR kód s MAC adresou nalepený na akčním členu.

V tento moment je možné se připojit k akčnímu členu. Pro připojení byla naprogramovaná třída ConnectingToBT.class. Při vytvoření tohoto objektu je potřeba v konstruktoru zadat UUID, který vytvoří BluetoothSocket. Ten po zavolání funkce *run()* zajistí spojení s Bluetooth modulem. V tento moment je možné již zahájit komunikaci a posílání dat mezi zařízeními. Následně je BluetoothSocket předán třídě ManageConnection.class, která zajišťuje správné odeslání dat.

V celém procesu připojování a komunikace může nastat řada výjimek, které je potřeba správně odchytit a ohlídat, aby nezpůsobily pád aplikace.

Ukázka funkce startConnecting(), která se stará o správné připojení k Bluetooth modulu a odchytení výjimek při selhání spojení:

```
public void startConnecting() {
    BluetoothDevice btDevice =
BluetoothAdapter.getDefaultAdapter().getRemoteDevice(user.getOurDeviceList(
).get(positionInDeviceList).getMacAddress());
    bluetoothConnect = new ConnectingToBT(btDevice);
    bluetoothConnect.start();
    Log.d(TAG, "startConnecting: ");
    new Thread(() -> {
        for (int i = 0; i <= 50; i++) {
            Log.d(TAG, "Connecting: " + i);
            if (bluetoothConnect.isConnected()) {
                runOnUiThread(() -> {
                    mDialog.dismiss();
                    isDeviceSet();
                });
                return;
            }
            try {
                Thread.sleep(100);
            } catch (InterruptedException e) {
                e.printStackTrace();
            }
        }
        runOnUiThread(() -> {
            mDialog.dismiss();
            AlertDialog.Builder builder = new AlertDialog.Builder(this);
            builder.setTitle("\Oj\");
            builder.setMessage("Nepodařilo se připojit k Bluetooth
zařízení, prosím zkontrolujte jestli je zapnuté.");
            builder.setPositiveButton("Ok", ((dialog, which) -> finish()));
            builder.setNegativeButton("Zrušit", ((dialog, which) -> finish()));
            mDialog = builder.create();
            mDialog.show();
        });
    }).start();
}
```

3.2.4. Odesílání dat

Pokud jsou dvě zařízení již připojena pomocí Bluetooth samotné odesílání dat je již jednoduché. V našem případě k němu byla vytvořena třída *ManageConnection.class*, která obsahuje veškeré potřebné funkce k odesílání dat.

Pro odeslání dat byly vytvořeny dvě funkce první je write(), která odesílá data pouze jednou za 20ms. Při tažení posuvníkem dochází ke generování velkého množství dat a pokud bychom je všechny nepřetržitě odesílali došlo by k zahlcení spojení. Druhou funkcí je writeAbsolutely(), která naopak zapíše okamžitě veškeré data které do ní vložíme. Použije se pouze pokud potřebujeme vědět, že hodnota byla skutečně odeslána.

Data je potřeba vložit jako řetězec bajtů, následně dochází k postupnému odesílání řetězce. Na druhé straně zařízení vytváří buffer, do kterého se bity zapisují. Po naplnění tohoto bufferu jsou data

dále zpracovány. V tomto případě by pro nás byl přenos dat nespolehlivý, jelikož by docházelo vzhledem k bufferu k latenci a také se objevil problém, že při rychlém zaslání docházelo k přetečení bufferu. Proto je v našem případě nutné od sebe zprávy oddělit. K oddělení jednotlivých zpráv použijeme jednoduchého symbolu `\n`, který v textovém řetězci značí konec řádku.

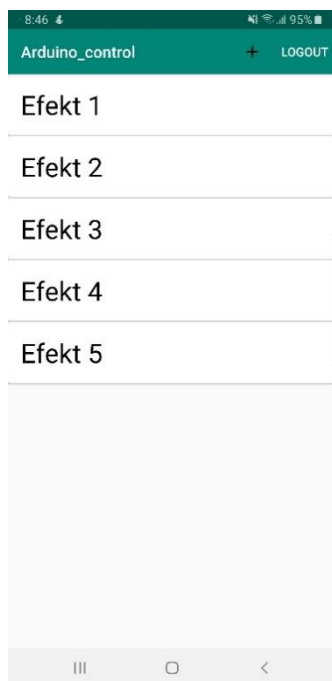
Při inicializaci objektu vnitřní třídy *ManageConnection.class* její konstruktor vytvoří objekt `OutputStream` do které bude odesílat veškerá data jež do něj zapíšeme pomocí funkce `write()` viz ukázka kódu:

```
public void write(byte[] bytes) {
    if (!sending) {
        try {
            sending = true;
            mmOutputStream.write(bytes);
            Log.d(TAG, "send: " + data);
            new Thread(() -> {
                try {
                    Thread.sleep(20);
                    sending = false;
                } catch (InterruptedException e) {
                    Log.d(TAG, "write: " + e);
                }
            }).start();
        } catch (Exception e) {
            Log.e(TAG, "Could not send data to device." + data);
            if (mmSocket.isConnected()) {
                bluetoothConnect.cancel();
            }
            startConnecting();
        }
    }
}
```

3.2.5. Třída *MainActivity.class*

Třída *MainActivity* je hlavní třídou, které je předáno řízení aplikace po přihlášení uživatele. Tato třída obsahuje grafické uživatelské rozhraní. Třída načte uživatelská data a zobrazí list uložených efektů, zároveň v pravém rohu, viz obr.26, zobrazí menu s možností odhlášení z aplikace a přidání dalšího efektu.

Při krátkém kliknutí na položku v listu efektů je předáno řízení třídě *BtControl*, která zahájí připojování k akčnímu členu. Naopak při dlouhém kliknutí cca 1 s je zobrazen dialog pro úpravu dané položky. Je možné změnit název nebo efekt odstranit.

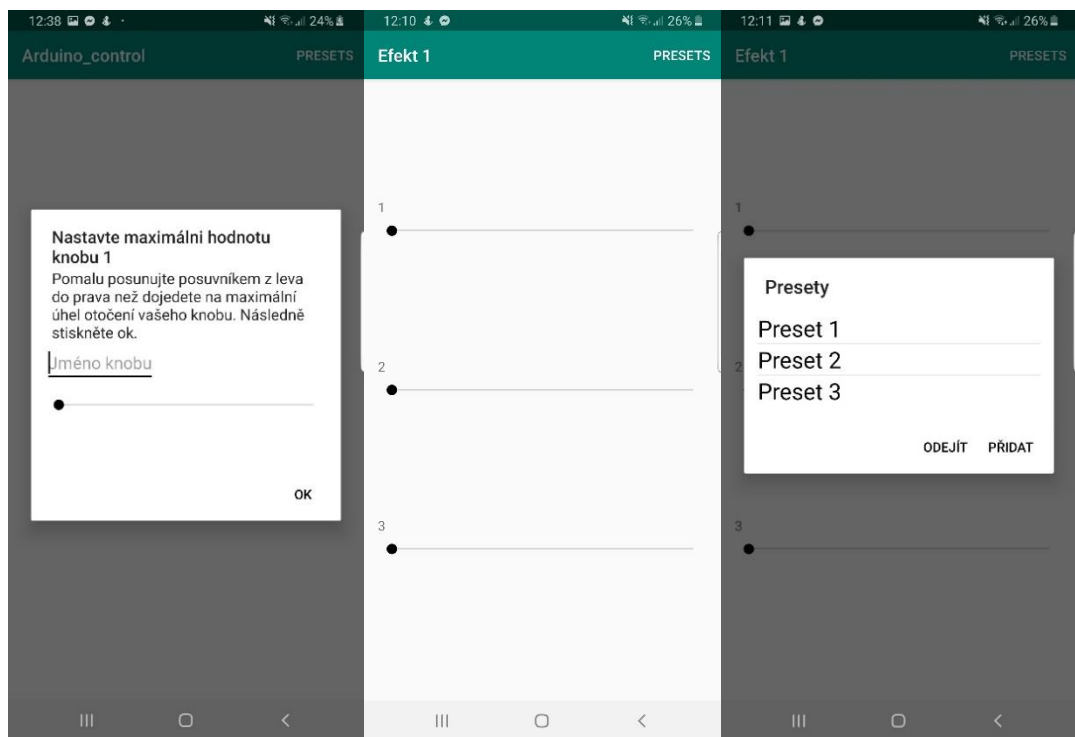


Obr. 26: Ukázka MainActivity

3.2.6. Třída BtControl.class

Třída *BtControl* byla vytvořena pro správu a řízení akčního členu spolu s Bluetooth komunikací. Obsahuje další dvě třídy: *ConnectingToBT.class* a *ManageConnection.class*. Díky této implementaci můžeme jednoduše vytvořit dva objekty, jež si mezi sebou předávají komunikační socket popsany v kapitole 3.2.4.

V momentě, kdy třída přebere řízení aplikace od třídy *MainActivity* obdrží od ní i UUID akčního členu ke kterému okamžitě zahájí připojování. Pokud se připojení podaří zobrazí uživateli ovládací prvky pro akční člen. Jde o 3 posuvníky kterými je možné ovládat parametry efektu. V pravém horním rohu, viz obr. 27, je menu pro spuštění dialogu s nastavením presetů, ve kterém je můžeme ukládat a vyvolávat. Pokud se uživatel vrátí zpět na *MainActivity* nebo zavře aplikaci dojde k automatickému odpojení od akčního členu, aby byla zajištěna nižší spotřeba akčního členu. Po návratu se aplikace ihned automaticky připojí k akčnímu členu.



Obr. 27: Ukázka BluetoothControlActivity

4. Cíl 3: Konstrukce a testování

4.1. Realizace akčního členu

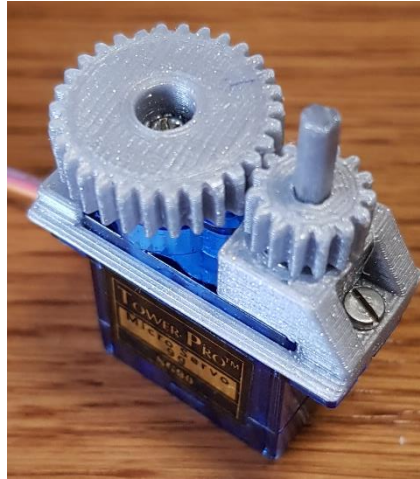
V této části bakalářské práce je pojednáno o sestrojení prototypu akčního členu. Většina konstrukce se navrhla tak, aby byla vytisknutelná na MDF 3D tiskárně. Pro tisk je použit plast PLA, který je sice měkký ale je levný a snadno tisknutelný. Pro prototyp akčního členu je dostačující.

Deska plošných spojů je realizována pomocí univerzální desky a propojení jsou řešena kabely.

4.1.1. Realizace uchycení servomotoru a převodu

Servomotor je přišroubován pomocí dvou M2 šroubů o délce 8 mm k držáku na servomotor. Tento držák zároveň obsahuje místo na uložení ložisek o rozměru 3 x 7 x 3 mm. Hnané ozubené kolo převodu je nasazeno a přišroubováno přímo na servomotor. Druhé hnací ozubené kolo převodu je přes hřídel nasazeno do ložiska.

Z důvodu nemožnosti vytisknout menší ozubená kola, bylo potřeba šetřit místem a druhé hnací ozubené kolo je přes hřídel připevněno k držáku servomotoru pouze na jedné straně, dochází k vyhnutí hřídele a následného přeskokování zubů v převodu. Tento problém byl částečně odstraněn zvednutím uložení ložiska, přidáním ještě jednoho tak, aby hnací ozubené kolo bylo umístěno co nejbližší k ložisku. Díky tomu se zamezí tak velkému vyhýbání hřídele.

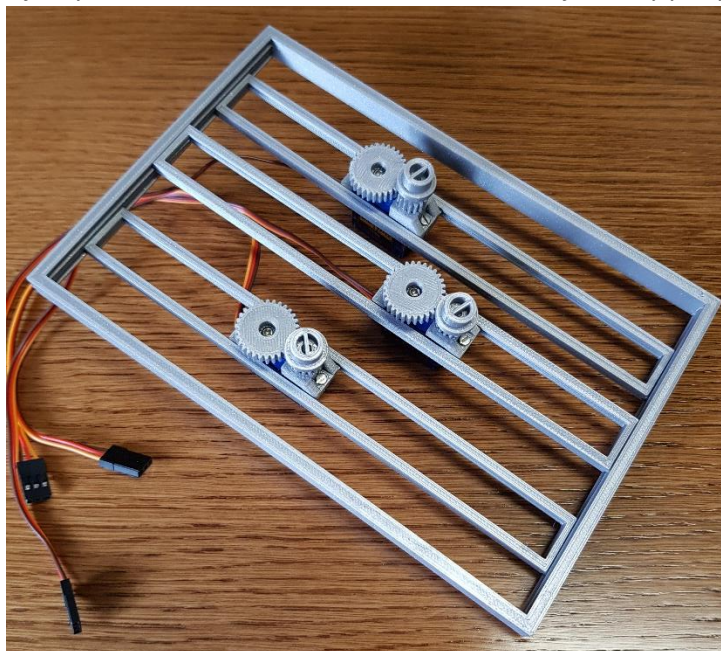


Obr. 28: Převod

4.1.2. Rám konstrukce

Při 3D tisku rámu konstrukce vznikl problém s kroucením plastu, takže se rám vytisknul prohnutý. Tento problém se podařilo vyřešit řádným přilepením rámu k tiskové podložce.

Rám se skládá ze 2 dílů. Z vlastního rámu, do kterého jsou vloženy pojezdy servomotoru. Tyto pojezdy však z rámu díky své délce vypadávají. Proto bylo nutné do rámu přidat hlazené pojezdové tyče. Tyto tyče drží pojezdy servomotorů v rámu a zároveň umožňují hladký pohyb viz bod 2.2.4.



Obr. 29 konstrukce akčního členu

4.2. Realizace řídící krabičky

V programu Solidworks byl navržen model krabičky přímo na míru použitému hardware. Obsahuje v sobě místo pro baterie, desku plošných spojů, Bluetooth modul a potřebné otvory pro konektory na servomotory, vypínač a usb vstup Arduina. Fyzická realizace krabičky je popsána níže.

4.2.1. Schéma zapojení

Pro šetření energie baterií je přidán vypínač napájení. Jeho zapojení je zobrazeno na obr. 30. Dále bylo potřeba vyřešit měření napětí baterie. Jak bylo popsáno v kapitole 3.1.3. Arduino měří napětí pomocí porovnání komparátorem s referenčním napětí 5 V. Použité baterie jsou zapojeny sériově a jejich napětí se díky tomu sčítá tedy 7,2 V. Aby bylo možné napětí baterie úspěšně měřit je napětí z baterie přivedeno na pin A1 přes dělič napětí.

Výpočet napěťového děliče:

Pro výpočet opět považujeme napěťový dělič jako nezatížený a je potřeba dosáhnout na výstupu polovičního napětí.

$$U_1 = 7,2 \text{ V}$$

$$U_2 = \frac{U_1}{2}$$

$$I = \frac{U_1}{R_1 + R_2}$$

$$U_2 = U_{R2} = I * R_2$$

$$U_2 = \frac{U_1}{R_1 + R_2} * R_2$$

$$U_2 = U_1 * \frac{R_2}{R_1 + R_2}$$

Z odvozeného vztahu plyne, že polovičního napětí docílíme, pokud budou oba odpory stejně velké. Volí se R_1 a R_2 1 k Ω . Pro kontrolu dosadíme do odvozeného vzorce.

$$U_2 = 7,2 * \frac{1000}{1000 + 1000} = 3,6 \text{ V}$$

Následně je potřeba zapojit kontrolní led diodu stavu baterie přes odpor obr. 30, aby nedocházelo k proudovému přetížení výstupního pinu desky Arduina. Pokud by nebyl pro připojení led diody použit odpor, tak by diodou protékal příliš velký proud, který by snižoval životnost led diody a zahříval malinké kontakty tranzistoru integrovaného v čipu až by se mohly odpařit.

Výpočet odporu:

V dokumentaci k led diodě je udáno maximální napětí 2,5 V a proud 20 mA. S Pro naše použití dioda dosáhne dostatečné svítivosti i při menším proudu. Aby byla šetřena výdrž baterie a nedocházelo ke zbytečnému zatěžování výstupních pinů arduina je zvoleno požadované napětí na led diodě 2,1 V a proud 10 mA.

$$U_0 = 5 \text{ V}, U_d = 2,1 \text{ V}, I = I_d = 10 \text{ mA}$$

$$U_0 = U_R + U_d$$

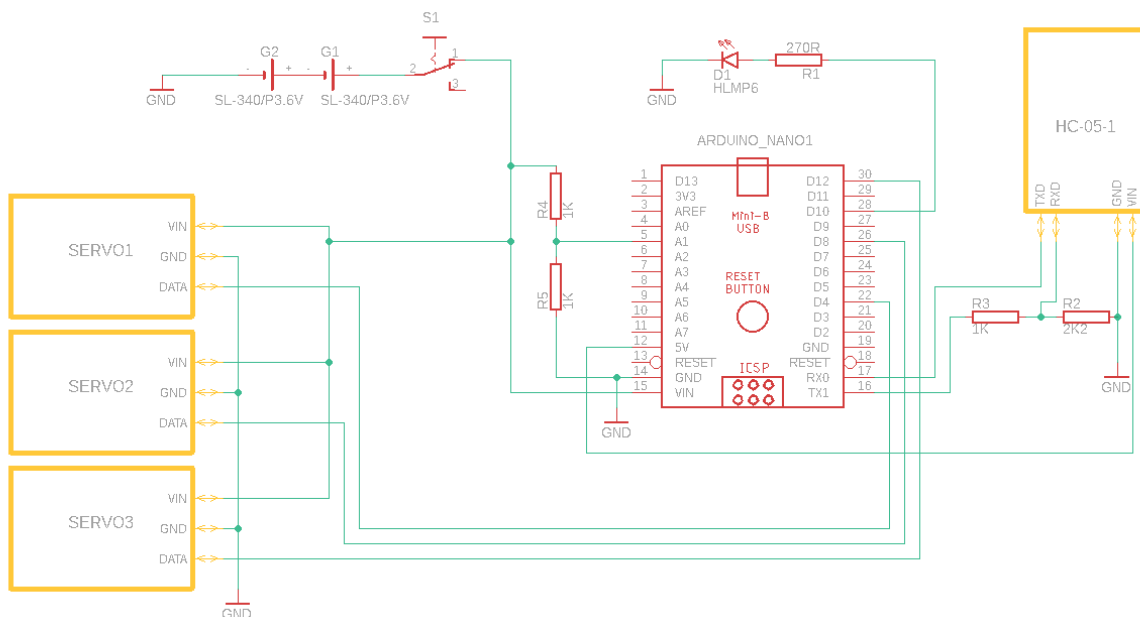
$$U_R = U_0 - U_d$$

$$U_R = 5 - 2,1 = 2,9 \text{ V}$$

$$R = \frac{U_R}{I}$$

$$R = \frac{2,9}{10 * 10^{-3}} = 290 \Omega$$

Rezistor volíme nejbližší nižší dle řady E24 270 Ω .



Obr. 30: Schéma zapojení

4.2.2. Držák na baterie

V modelu krabičky bylo přesně počítáno s umístěním baterií. Do vytištěné plastové krabičky se na své místa umístily bateriové kontakty. Na jedné straně byly kontakty spájeny k sobě obr. 31 a na straně druhé jsou vyvedeny jako plus a mínus do části pro elektroniku.

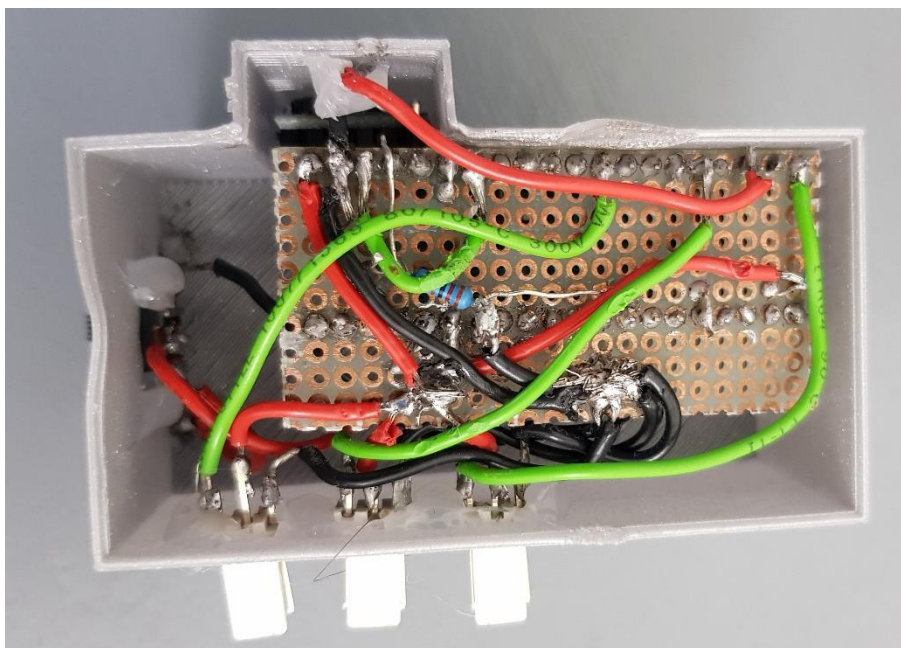


Obr. 31: Držák na baterie

4.2.3. Realizace desky plošných spojů

Jako základ desky plošných spojů byl použit univerzální vrtaný plošný spoj s kulatými body. Do této desky se následně osadila dutinková lišta pro odnímatelné osazení arduina. Do lišty je také zapojen Bluetooth modul.

Následné propojení s konektory pro servomotory, napájení a s indikační led diodou je realizováno pomocí kabelových propojek nebo přímo připojovacími vodiči rezistorů.



Obr. 32: Deska plošných spojů osazena v krabičce

5. Nápady na vylepšení

5.3. Použití BLE

U prototypu byl použit Bluetooth modul HC-05, který podporuje starý protokol bluetooth classic v2. Novější protokoly umožňují komunikaci Bluetooth low energy, zkráceně BLE, pro nízkou spotřebu energie a jednodušší práci. Nevýhodou je pomalá rychlost přenosu dat, maximálně 2Mbps. Se zvyšující vzdáleností rychlost klesá až na 125kbps. [23]

Tato rychlost přenosu je však pro tento koncept dostačující. BLE má již v sobě zabudovaný například mikropočítač ESP32. Takto by bylo docíleno menší spotřeby baterie. Bluetooth classic 2.0 má spotřebu cca 30 – 50 mA zatím co BLE 5.0 má spotřebu menší než 15 mA.

5.4. Vytvoření webové aplikace

Správu BLE podporují i webové prohlížeče například Chrome. Díky tomu by bylo možné vytvořit webovou aplikaci, která se přímo spojí s akčním členem.

Bylo by díky tomu docíleno kompatibility se všemi zařízeními obsahující Bluetooth a webový prohlížeč. Také by pro všechny platformy, zejména IOS, Android, Windows stačila jediná webová aplikace.

U prototypu je již navrhnutá databáze pomocí služby Firebase tato databáze je jednoduše přenositelná do webové aplikace. Uživatel by mohl střídat jakékoliv zařízení a měl by vše dostupnou stejnou aplikaci se svými daty jednoduše ve webovém prohlížeči.

5.5. Uchycení akčního členu

V průběhu bakalářské práce byl nalezen podobný systém zmíněný v kapitole 1.2.5. Tento systém řeší uchycení akčního členu přímo k potenciometru pomocí pojistné matice.

Použitím tohoto stylu upevnění by odpadla potřeba rámu pro servomotory, díky tomu se celý systém stane jednodušším. Zvýší se jeho univerzálnost a sníží náklady na výrobu.

5.6. Ovládání na akčním členu

Prototyp z důvodu jednoduchosti řešení neumožňuje ovládání parametru efektu přímo z akčního členu. Pokud je nasazen na efektu, lze jeho parametry ovládat pouze přes aplikaci.

Tuto nevýhodu by šlo odstranit přidáním modulu rotačního enkodéru ke každému servomotoru. Díky tomu by bylo možné parametry ovládat i přímo z akčního členu.

5.7. Ovládání přes MIDI protokol

Populární protokol pro ovládání hudebních efektů popsán v kapitole 1.2.3. Implementaci MIDI ovládání by šlo realizovat pomocí přídavného modulu, který zpracuje MIDI příkazy, tak aby s nimi mohl mikropočítač pracovat dál.

Implementací by se dosáhlo zvýšení univerzálnosti a rozšíření možností systému například o řízení akčního členu pomocí automatizací v DAW. Díky tomu by se mohl akční člen sám nastavovat v průběhu písničky dle předchozího nastavení.

ZÁVĚR

V rámci Bakalářské práce byl stanoven cíl potvrdit funkčnost konceptu centralizovaného ovládání hudebních efektů. Tento koncept se v rámci práce podařilo potvrdit vytvořením prototypu akčního členu spolu s potřebnou IT infrastrukturou pro jeho řízení. Bakalářská práce byla rozdělena do 3 podcílů.

Prvním byl návrh konstrukce akčního členu. Návrh proběhl v programu Solidworks a při práci nastaly problémy zejména při navrhování konstrukčních částí tak, aby byly tisknutelné metodou MDF. 3D tisk má poměrně velké tolerance u použité tiskárny cca +- 0,1mm. Části akčního členu jsou malé, aby se vešly na efekt vedle sebe a velké tolerance 3D tisku díky tomu ztěžovaly návrh konstrukce. Bylo nutné průběžně testovat, jestli navržené díly skutečně sedí.

Druhým cílem bylo vytvoření IT infrastruktury. V rámci toho cíle byla vytvořena databáze pomocí dostupného řešení Firebase, mobilní aplikace pro platformu Android obsahující uživatelské rozhraní pro ovládání akčního členu ukládání přednastavení a obstarávající komunikaci s akčním členem. Jako poslední byl vytvořen řídicí program pro Arduino, který se stará o přijímání dat přes Bluetooth a jejich převod do úhlu natočení servomotorů. Při realizaci infrastruktury bylo zjištěno, že pro toto použití by bylo mnohem vhodnější využití protokolu Bluetooth low energy, jež je popsán v kapitole 5.3.

Třetím cílem byla funkční realizace. Podařilo se tedy vytvořit funkční akční člen který jde ovládat přes mobilní aplikaci. Bohužel díky použití 3D tisku bylo konstrukční řešení limitované velikostí a nebylo možné sestavit vhodnější převod. Díky tomu dochází k občasnému přeskokování mezi zuby ozubených kol. Tento problém je popsán v kapitole 4.1.1.

Práci shledávám úspěšnou a přínosnou. Byly zjištěny problémy při realizaci konceptu a navrženy vhodné řešení pro jejich odstranění a rozšíření funkčnosti konceptu

SEZNAM ZDROJŮ

- [1] PTÁČEK Ladislav, Jak pracují efekty I, [online].2009, [cit. 21.5.2021], Dostupné z URL: <<http://www.muzikus.cz/pro-muzikanty-clanky/Jak-pracuji-efekty-I~19~leden~2009/>>
- [2] Vox, AC30, [online], [cit. 21.5.2021], Dostupné z URL: <<https://voxamps.com/product/ac30-custom-head/>>
- [3] World Headquarters, HeadRush Pedalboard, HEADRUSH, [online], [cit. 21.12.2019], dostupné z URL: <<https://www.headrushfx.com/pedalboard-pedalboardxus#>>
- [4] HANNES Bieger, Question de son, [online].2018, [cit. 21.5.2021], Dostupné z URL: <<https://www.soundonsound.com/music-business/question-de-son-paris>>
- [5] DISK MULTIMEDIA, Jak využít moduly API 500?, [online].2015, [cit. 10.12.2019], Dostupné z URL: <<https://magazin.disk.cz/cs/jak-vyuzit-moduly-api-500>>
- [6] AUTOMATED PROCESSES, 6 Slot high current lunchbox., APIAUDIO, [online], [cit.12.12.2019], Dostupné z URL: <<http://apiaudio.com/product.php?id=109>>
- [7] REVERB, Beginner`s guide to Eurorack, [online].2020, [cit. 21.5.2021], Dostupné z URL: <<https://reverb.com/news/beginners-guide-to-eurorack-case-basics-oscillators-filters>>
- [8] BRODY Mark, Gear Trends, [online].2018, [cit. 21.5.2021], Dostupné z URL: <<https://www.sweetwater.com/insync/the-explosion-of-eurorack/>>
- [9] FADERFOX, MX12 – Mix controller, [online], [cit. 22.5.2021], Dostupné z URL: <<http://www.faderfox.de/mx12.html>>
- [10] VODOO LAB, Ground control pro, [online], [cit. 22.5.2021] Dostupné z URL: <<http://www.voodoolab.com/gcontrolpro.htm>>
- [11] TOENGI Thomas, Geckotool, [online].2019 [cit. 22.5.2021] Dostupné z URL: <<https://www.kickstarter.com/projects/geckotool/geckotool-total-control-over-your-music-gear>>
- [12] CREATIVE COMMONS, Co je to Arduino?, [online], [cit. 10.12.2019], Dostupné z URL: <<https://arduino.cz/co-je-to-arduino/>>[Attribution-ShareAlike 3.0 Unported(CC BY-SA 3.0)]
- [13] ARDUINO, Arduino NANO [online], [cit. 23.5.2021], Dostupné z URL: <<https://store.arduino.cc/arduino-nano>>
- [14] SAKUL, Bluetooth HC-05, Lukáš Kořínek, [online].2017, [cit. 12.12.2019], dostupné z URL: <<http://www.sakul.cz/bluetooth-hc-05/n>>
- [15] ING. POSPÍŠIL Jaroslav, Elektrotechnika 1, [online], [cit. 23.5.2021], Dostupné z URL: <<https://www.vovcr.cz/odz/tech/381/page18.html>>
- [16] ELEKTRONIKA JE COOL, Cyklus For a PWM, [online], [cit. 12.12.2019], Dostupné z URL: <<http://elektronika.jecool.net/jas-led-cyklus-for-pwm/>>
- [17] Laskarduino, micro servo SG90, [online], [cit. 23.5.2021], Dostupné z URL: <<https://www.laskarduino.cz/plastove-micro-servo-sg90-9g--180/>>

[18] ING. KRATOCHVÍL Michal, Eluc, [online].2015, [cit. 24.5.2021], Dostupné z URL:
<<https://eluc.kr-olomoucky.cz/verejne/lekce/1904>>

[19] ING. KLAPKA Milan, Konstruování strojů, [online], [cit. 24.5.2021], Dostupné z URL:
<https://kke.zcu.cz/export/sites/kke/about/projekty/enazp/projekty/02_Silnicni-vozidla_4-6/5_MMP/004_Ozubena-kola---Klapka---P1.pdf>

[21] FIREBASE, Documentation, [online], [cit. 27.5.2021], Dostupné z URL:
<<https://firebase.google.com/docs>>

[20] CREATIVE COMMONS, Arduino, [online].2019, [cit. 10.12.2019], Dostupné z URL:
<<https://www.arduino.cc/en/reference/servo>>
< <https://www.arduino.cc/reference/en/libraries/servo/>>

[22] GOOGLE DEVELOPERS, Build anything on Android, [online], [cit. 21.12.2019], dostupné z URL:
<<https://developer.android.com/reference/android/content/Intent>> pro bod 4.6.
<<https://developer.android.com/guide/topics/connectivity/bluetooth>>pro bod 4.5.2

[23] NGUYEN Augustine, Symmetry Electronics, [online].2018, [cit. 30.5.2021], dostupné z URL:
<<https://www.semiconductorstore.com/blog/2018/Bluetooth-1-0-vs-2-0-vs-3-0-vs-4-0-vs-5-0-How-They-Differ-Symmetry-Blog/3147/>>

SEZNAM OBRÁZKŮ

- Obr. 1: Kytarová hlava Vox AC 30 [2]*
- Obr. 2: Analogový kytarový delay efekt*
- Obr. 3: Multiefektový procesor HEADRUSH [3]*
- Obr. 4: Studiový rack s efekty [4]*
- Obr. 5: Ukázka VST pluginu RoomWorks SE v DAW softwaru Cubase*
- Obr. 6: Api lunchbox [6]*
- Obr. 7: Eurorack [8]*
- Obr. 8 MIDI kontroler [9]*
- Obr. 9: Geckotool [11]*
- Obr. 10: Arduino NANO [13]*
- Obr. 11: Zapojení Bluetooth modulu HC-05 [14]*
- Obr. 12: Návrh nezatíženého děliče-výpočet rezistorů [15]*
- Obr. 13: Pulzně šířková modulace [16]*
- Obr. 14: Servo SG90 [17]*
- Obr. 15: Li-ion baterie*
- Obr. 16: Uchycení servomotoru*
- Obr. 17: Adaptér na potenciometr*
- Obr. 18 a 19: Sestava akčního členu*
- Obr. 20: Krabíčka pro elektroniku*
- Obr. 21: Vývojové prostředí Arduino IDE*
- Obr. 22: Blokový diagram programu pro Arduino*
- Obr. 23: Vývojové prostředí Android studio*
- Obr. 24 a 25: Blokové schéma aplikace pro Android*
- Obr. 26: Ukázka Main activity*
- Obr. 27: Ukázka BluetoothControl activity*
- Obr. 28: Převod*
- Obr. 29 konstrukce akčního členu*
- Obr. 30: Schéma zapojení*
- Obr. 31: Držák na baterie*
- Obr. 32: Deska plošných spojů osazena v krabíčce*

SEZNAM SYMBOLŮ, VELIČIN A ZKRATEK

Zkratky:

DSP	...	digital signal processor (digitální signálový procesor)
DAW	...	digital audio workstation (digitální pracovní audio rozhraní)
VST	...	virtuál studio technology (virtuální studiová technologie)
CPU	...	central processing unit (centrální procesorová jednotka)
PWM	...	pulse width modulation (pulzně šířková modulace)
UUID	...	universally unique identifier (univerzální unikátní identifikátor)
BLE	...	Bluetooth low energy

Veličiny:

U	...	napětí	[V]
I	...	proud	[A]
R	...	odpor	[Ω]
i	...	převodový poměr	[-]
Z	...	počet zubů ozubeného kola	[-]
D	...	průměr ozubeného kola	[m]
m	...	modul ozubeného kola	[m]

SEZNAM PŘÍLOH

A. Obsah přiloženého multimédia

A. Obsah přiloženého multimédia

- Bakalářská práce v elektronické podobě
- Složka souborů „Obrázky“ obsahující obrázky použity v bakalářské práci
- Složka souborů „Video“ obsahující video funkčního prototypu
- Složka souborů „Solidworks“ obsahující všechny navržené díly ve formátu SLDPRT
- Složka souborů „Schéma“ obsahující schéma zapojení navržené v programu EAGEL
- Složka souborů „Arduiono“ obsahující zdrojový kód pro řízení Arduina
- Složka souborů „Android“ obsahující zdrojové kódy mobilní aplikace pro platformu android